AMENDMENTS IN THE CLAIMS

1. (Previously Presented) A method for generating a primary scrambling code, the method comprising the steps of:

generating a first m-sequence from a first m-sequence generator including first shift registers having first shift register values a_i , wherein i = 0 to c-1 and where c is the total number of the registers;

generating a second m-sequence from a second m-sequence generator including second shift registers having values b_j , wherein j=0 to c-1, and where c is the total number of the registers;

masking the first shift register values a_i with a first set of mask values K_i , wherein i = 0 to c-1 to generate a third m-sequence;

adding the first m-sequence with the second m-sequence to generate a primary scrambling code; and

adding the third m-sequence and the second m-sequence to generate a secondary scrambling code;

wherein, the masking step shifts the first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code.

2-20. (Cancelled)

21. (Previously Presented) A scrambling code generator, comprising:

a first m-sequence generator to generate a first m-sequence by using a plurality of first registers with first shift register values a_i , wherein i = 0 to c-1 and where c is the total number of the first registers;

a second m-sequence generator to generate a second m-sequence by using a plurality of second registers with second shift register values b_j , wherein j = 0 to c-1 and where c is the total number of second registers;

a masking section to mask the first shift register values a_i with a first set of mask values K_i to generate a third m-sequence, wherein i = 0 to c-1 to generate a third m-sequence;

a first adder to add the first m-sequence and the second m-sequence to generate a primary scrambling code; and

a second adder to add the third m-sequence and the second m-sequence to generate a secondary scrambling code,

wherein the masking section shifts the first m-sequence cyclically by L chips to generate an Lth secondary scrambling code associated with the primary scrambling code.

22-30. (Cancelled)

- 31. (Previously Presented) The method of claim 1, wherein the primary scrambling code is one of a plurality primary scrambling codes and a Kth primary scrambling code is a ((K-1)*M+K)th gold code, where M is a total number of secondary scrambling codes per primary scrambling code and 1<K<512.
- 32. (Previously Presented) The method of claim 1, wherein the secondary scrambling codes associated with a Kth primary scrambling code are from ((K-1)*M+K+1)th to (K*M+K)th gold codes, where M is a total number of secondary scrambling codes per primary scrambling code and 1<K<512.
- 33. (Previously Presented) The method of claim 1, wherein 1<L<M, where M is a total number of secondary scrambling codes per primary scrambling code.
- 34. (Previously Presented) The method of claim 1, wherein the masking step is expressed by $\sum (k_i \times a_i)$.
 - 35. (Previously Presented) The method of claim 1, further comprising:

masking the first shift register values a_i with a second set of mask values K_j to generate a fourth m-sequence, wherein j=0 to c-1; and

adding the fourth m-sequence and the second m-sequence to generate an N^{th} secondary scrambling code associated with the primary scrambling code;

wherein, the masking step shifts the first m-sequence cyclically by N chips to generate an N^{th} secondary scrambling code.

- 36. (Previously Presented) The method of claim 35, wherein 1<N<M, where M is a total number of secondary scrambling codes per primary scrambling code.
- 37. (Previously Presented) The method of claim 1, further comprising the step of delaying at least one of the primary scrambling code and secondary scrambling code to produce a Q-channel component, wherein the primary scrambling code and secondary scrambling code are I-channel components.
- 38. (Previously Presented) The scrambling code generator of claim 21, wherein the primary scrambling code is one of a plurality of primary scrambling codes and a Kth primary scrambling code is a ((K-1)*M+K)th gold code, where M is a total number of secondary scrambling codes per primary scrambling code and 1<K<512.
- 39. (Previously Presented) The scrambling code generator of claim 38, wherein the secondary scrambling codes associated with the Kth primary scrambling code are ((K-1)*M+K+1)th to (K*M+K)th gold codes.
- 40. (Previously Presented) The scrambling code generator of claim 21, further comprising:

a second masking section to mask the first shift register values a_i , with a second set of mask values K_j , wherein j = 0 to c-1, to generate a fourth m-sequence; and

a third adder to add the fourth m-sequence and the second m-sequence to generate an N-th secondary scrambling code associated with the primary scrambling code,

wherein the second masking section shifts the first m-sequence cyclically by N chips to generate the Nth secondary scrambling code.

- 41. (Previously Presented) The scrambling code generator of claim 21, wherein the masking section shifts the first m-sequence cyclically by masking the first shift register values a_i in accordance with $\sum (K_i \times a_i)$.
- 42. (Previously Presented) The scrambling code generator of claim 21, wherein the first m-sequence generator cyclically shifts the first shift register values and the second m-sequence generator cyclically shifts the second shift register values.
- 43. (Previously Presented) The scrambling code generator of claim 21, wherein the first m-sequence generator adds predetermined shift register values of the first shift registers based on a first generating polynomial of the first m-sequence, right shifts the first shift register values a_i of the first shift registers, and replaces the first register value a_{c-1} with the result of the addition of the predetermined register values.
- 44. (Previously Presented) The scrambling code generator of claim 21, wherein the first m-sequence generator adds a first shift register value a_0 with a first shift register a_7 to form a next first shift register a_{c-1} .
- 45. (Previously Presented) The scrambling code generator of claim 21, wherein the second m-sequence generator adds predetermined shift register values of the second shift registers based on a second generating polynomial of the second m-sequence, right shifts the second shift register values b_j of the second shift registers, and replaces the second register value b_{c-1} with the result of the addition of the predetermined register values.
- 46. (Previously Presented) The scrambling code generator of claim 21, wherein the second m-sequence generator adds a second shift register value b_0 with a second shift register value b_5 , b_7 , and a second shift register value b_{10} to form a next second shift register value b_{c-1} .
- 47. (Previously Presented) The apparatus of claim 21, further comprising a means for delaying at least one of the primary scrambling code and the secondary scrambling code to

produce Q-channel component, wherein the primary scrambling code and the secondary scrambling code are I-channel components.

48 - 53. (Cancelled)

54. (Currently Amended) A method for generating scrambling codes in mobile communication system having a scrambling code generator, the method comprising steps of:

generating a ((K-1)*M+K)th gold code as a Kth primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code; and

generating $((K-1)*M+K+1)^{th}$ through $(K*M+K)^{th}$ gold codes as secondary scrambling codes associated with the K^{th} primary scrambling code,

wherein an the Lth Gold code is generated by adding an (L-1)-times shifted first m-sequence and a second m-sequence.

- 55. (Previously Presented) The method as claimed in claim 54, wherein K is a primary scrambling code number and $1 \le K \le 512$.
- 56. (Previously Presented) The method as claimed in claim 55, wherein the first m-sequence is generated from a first shift register memory having a plurality of first shift registers with first shift register values a_i , wherein i = 0 to c-1 and where c is the total number of the first registers and the (L-1)-times shifted first m-sequence is generated by masking the first shift register values a_i with mask values K_i , where i = 0 to c-1.
- 57. (Previously Presented) The method as claimed in claim 56, wherein the masking is performed according to: $\sum (K_i \times a_i)$.
- 58. (Previously Presented) The method as claimed in claim 54, wherein the generated primary scrambling code and secondary scrambling code are I-channel components and the

method further comprises delaying at least one of the primary scrambling code and secondary scrambling code to produce Q-channel components.

- 59. (Previously Presented) An apparatus for generating scrambling codes in mobile communication system having a scrambling code generator, comprising:
 - a first m-sequence generator to generate a first m-sequence;
 - a second m-sequence generator to generate a second m-sequence; and
- at least one adder for generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence,

wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

- 60. (Previously Presented) The apparatus of claim 59, wherein the secondary scrambling codes of the Kth primary scrambling codes are the ((K-1)*M+K+1)th through (K*M+K)th Gold codes.
- 61. (Previously Presented) The apparatus as claimed in claim 60, wherein K is a primary scrambling code number and $1 \le K \le 512$.
- 62. (Previously Presented) The apparatus as claimed in claim 59, wherein the first m-sequence generator comprises a plurality of first registers with first shift register values a_i , wherein i = 0 to c-1 and where c is the total number of the first shift registers, and the scrambling code generator further comprising at least one masking section for generating the n-times shifted first m-sequence by masking the first shift register values a_i with mask values K_i , where i = 0 to c-1.
- 63. (Previously Presented) The apparatus as claimed in claim 62, wherein the masking is performed according to: $\sum (K_i \times a_i)$.

- 64. (Previously Presented) The apparatus as claimed in claim 59, wherein the primary scrambling code and secondary scrambling code are I-channel components and the apparatus further comprises a means for delaying at least one of the primary scrambling codes and secondary scrambling code to produce Q-channel components.
- 65. (Previously Presented) A method for generating scrambling codes in mobile communication system having a scrambling code generator, comprising the steps of:

generating a first m-sequence;

generating a second m-sequence; and

generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence,

wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

- 66. (Previously Presented) The method as claimed in claim 65, further comprising generating ((K-1)*M+K+1)th to (K*M+K)th Gold codes as secondary scrambling codes corresponding to the Kth primary scrambling code.
- 67. (Previously Presented) The method as claimed in claim 65, wherein K is a primary scrambling code number and $1 \le K \le 512$.
- 68. (Previously Presented) The method as claimed in claim 65, wherein the first m-sequence is generated from a first shift register memory having a plurality of first shift registers with first shift register values a_i , wherein i = 0 to c-1 and where c is the total number of the first registers and the n-times shifted first m-sequence is generated by masking the first shift register values a_i with mask values K_i , where i = 0 to c-1.
- 69. (Previously Presented) The method as claimed in claim 68, wherein the masking is performed according to: $\sum (K_i \times a_i)$.

70. (Previously Presented) The method as claimed in claim 65, wherein each scrambling code is used as an I-channel component and a Q-channel component, corresponding to the I-channel component, is generated by delaying the I-channel component for a predetermined time.